What is claimed is:

1. In a data-packet processor, a configurable queueing system for packet accounting during processing, the system comprising:

a plurality of queues arranged in one or more clusters;

an identification mechanism for creating a packet identifier for arriving packets;

insertion logic for inserting packet identifiers into queues and for determining into which queue to insert a packet identifier; and

selection logic for selecting packet identifiers from queues to initiate processing of identified packets, downloading of completed packets, or for requeueing of the selected packet identifiers.

2. The queueing system of claim 1 wherein the total number of packet identifiers in all queues is fixed and the number of clusters and the number of queues per cluster may vary from 1 to a same maximum number, such that with the number of clusters equal one the number of queues is the maximum number and vice-versa.

3. The queueing system of claim 2 wherein the relative number of clusters, determining the number of queues, is set by setting a configuration register associated with the processor.

4. The queueing system of claim 2 wherein the total number of packet identifiers is 256, meaning that 256 packets may be in process at any moment in time in the data packet processor.

5. The queueing system of claim 2 wherein the maximum number of either clusters or queues is 32, and the number of clusters or queues is controlled to be a power of 2.

6. The queueing system of claim 1 wherein the packet identifier created is associated to a memory location where the packet itself is stored during processing.

7. The queueing system of claim 6 wherein an on-board memory for storage of packets data during processing is arranged in packet pages, and the packet identifier is an address identifying the beginning of a packet page where the packet data for a packet is stored.

8. The queueing system of claim 6 wherein the packet identifier created and queued comprises, in addition to the memory location, one or more of an identifier of a device from which the packet was received, an indication of whether the packet has increased or decreased in total size during processing, an indication of whether or not processing is completed on the associated packet, a next queue into which the packet identifier should be inserted, and an identifier of whether the packet is currently being processed.

9. The queueing system of claim 1 wherein the insertion logic determines a priority for a packet, and inserts the packet into a particular queue depending on the priority determined.

10. The queueing system of claim 9 wherein clusters are numbered, and priority for processing identified packets is by cluster number.

11. The queueing system of claim 8 wherein the selection system selects packets for requeueing based on the value of next queue number set in the packet identifier.

12. The queueing system of claim 8 wherein the selection system selects packets for downloading based on the value of the indication of whether processing on the associated packet is completed.

13. The queueing system of claim 10 wherein the selection system selects packets for processing based upon priority by queue number.

14. The queueing system of claim 13 wherein a fairness algorithm is followed in selecting a queue within a priority cluster for selecting a packet for processing.

15. The queueing system of claim 1 wherein the data-packet processor is a multi-streaming processor having separate contexts implemented by context registers and associated with individual streams, and the selection system causes header information from a packet associated with a selected identifier to be loaded into an available context.

16. The queueing system of claim 15 wherein there are more contexts than streams, and loading of header information into available contexts is accomplished in background, that is, in parallel with processing of packets.

17. The queueing system of claim 8 wherein software executed by the processor is enabled to manipulate value of individual ones of fields in a packet identifier.

18. The queueing system of claim 1 wherein the insertion logic inserts packets into queues at least in part according to flow, and wherein insertion and selection logic cooperate to preserve order of packets in individual flows.

19. The queueing system of claim 1 wherein the insertion logic treats packets incoming from the network or switching fabric with highest priority for identifier insertion into a queue, packet identifiers to be requeued with a next highest priority, and identifiers for packets generated by software with a lowest priority.

20. A data-packet processor, comprising:

an input buffer for buffering incoming packets from one or more sources;

a local packet memory for storing packet data during processing;

a stream processor unit for processing packet data; and

a configurable queueing system for packet accounting during processing;

wherein the queueing system comprises a plurality of queues arranged in one or more clusters, an identification mechanism for creating a packet identifier for arriving packets, insertion logic for inserting packet identifiers into queues and for determining into which queue to insert a packet identifier, and selection logic for selecting packet identifiers from queues to initiate processing of identified packets, downloading of completed packets, or for requeueing of the selected packet identifiers.

21. The processor of claim 20 wherein the total number of packet identifiers in all queues is fixed and the number of clusters and the number of queues per cluster may vary from 1 to a same maximum number, such that with the number of clusters equal one the number of queues is the maximum number and vice-versa.

22. The processor of claim 21 wherein the relative number of clusters, determining the number of queues, is set by setting a configuration register associated with the processor.

23. The processor of claim 21 wherein the total number of packet identifiers is 256, meaning that 256 packets may be in process at any moment in time in the data packet processor.

24. The processor of claim 21 wherein the maximum number of either clusters or queues is 32, and the number of clusters or queues is controlled to be a power of 2.

25. The processor of claim 20 wherein the packet identifier created is associated to a memory location where the packet itself is stored during processing.

26. The processor of claim 25 wherein an on-board memory for storage of packet data during processing is arranged in packet pages, and the packet identifier is an address identifying the beginning of a packet page where the packet data for a packet is stored.

27. The processor of claim 25 wherein the packet identifier created and queued comprises, in addition to the memory location, one or more of an identifier of a device from which the packet was received, an indication of whether the packet has increased or decreased in total size during processing, an indication of whether or not processing is completed on the associated packet, a next queue into which the packet identifier should be inserted, and an identifier of whether the packet is currently being processed.

28. The processor of claim 20 wherein the insertion logic determines a priority for a packet, and inserts the packet into a particular queue depending on the priority determined.

29. The processor of claim 28 wherein clusters are numbered, and priority for processing identified packets is by cluster number.

30. The processor of claim 27 wherein the selection system selects packets for requeueing based on the value of next queue number set in the packet identifier.

31. The processor of claim 27 wherein the selection system selects packets for downloading based on the value of the indication of whether processing on the associated packet is completed.

32. The processor of claim 29 wherein the selection system selects packets for processing based upon priority by queue number.

33. The processor of claim 32 wherein a fairness algorithm is followed in selecting a queue within a priority cluster for selecting a packet for processing.

34. The processor of claim 20 wherein the data-packet processor is a multi-streaming processor having separate contexts implemented by context registers and associated with individual streams, and the selection system causes header information from a packet associated with a selected identifier to be loaded into an available context.

35. The processor of claim 34 wherein there are more contexts than streams, and loading of header information into available contexts is accomplished in background, that is, in parallel with processing of packets.

36. The processor of claim 27 wherein software executed by the processor is enabled to manipulate value of individual ones of fields in a packet identifier.

37. The processor of claim 20 wherein the insertion logic inserts packets into queues at least in part according to flow, and wherein insertion and selection logic cooperate to preserve order of packets in individual flows.

38. The processor of claim 20 wherein the insertion logic treats packets incoming from the network or switching fabric with highest priority for identifier insertion into a queue, packet identifiers to be requeued with a next highest priority, and identifiers for packets generated by software with a lowest priority.

39. A method for accounting for packets in a data-packet processor, comprising the steps of:

(a) providing in the processor a plurality of queues in one or more clusters;

(b) creating a packet identifier for arriving packets;

(c) inserting packet identifiers into queues by an insertion mechanism having logic for determining into which queue to insert; and

(d) selecting packet identifiers from the queues by selection logic, for initiating processing of identified packets, downloading of completed packets, or for requeueing of the selected packet identifiers.

40. The method of claim 39 wherein the total number of packet identifiers in all queues is fixed and the number of clusters and the number of queues per cluster may vary from 1 to a same maximum number, such that with the number of clusters equal one the number of queues is the maximum number and vice-versa.

41. The method of claim 40 wherein the relative number of clusters, determining the number of queues, is set by setting a configuration register associated with the processor.

42. The method of claim 40 wherein the total number of packet identifiers is 256, meaning that 256 packets may be in process at any moment in time in the data packet processor.

43. The method of claim 40 wherein the maximum number of either clusters or queues is 32, and the number of clusters or queues is controlled to be a power of 2.

44. The method of claim 39 wherein the packet identifier created is associated to a memory location where the packet itself is stored during processing.

45. The method of claim 44 wherein an on-board memory for storage of packets data during processing is arranged in packet pages, and the packet identifier is an address identifying the beginning of a packet page where the packet data for a packet is stored.

46. The method of claim 44 wherein the packet identifier created and queued comprises, in addition to the memory location, one or more of an identifier of a device from which the packet was received, an indication of whether the packet has increased or decreased in total size during processing, an indication of whether or not processing is completed on the associated packet, a next queue into which the packet identifier should be inserted, and an identifier of whether the packet is currently being processed.

47. The method of claim 39 wherein the insertion logic determines a priority for a packet, and inserts the packet into a particular queue depending on the priority determined.

48. The method of claim 47 wherein clusters are numbered, and priority for processing identified packets is by cluster number.

49. The method of claim 46 wherein the selection system selects packets for requeueing based on the value of next queue number set in the packet identifier.

50. The method of claim 46 wherein the selection system selects packets for downloading based on the value of the indication of whether processing on the associated packet is completed.

51. The method of claim 48 wherein the selection system selects packets for processing based upon priority by queue number.

52. The method of claim 13 wherein a fairness algorithm is followed in selecting a queue within a priority cluster for selecting a packet for processing.

53. The method of claim 39 wherein the data-packet processor is a multi-streaming processor having separate contexts implemented by context registers and associated with individual streams, and the selection system causes header information from a packet associated with a selected identifier to be loaded into an available context.

54. The method of claim 53 wherein there are more contexts than streams, and loading of header information into available contexts is accomplished in background, that is, in parallel with processing of packets.

55. The method of claim 8 wherein software executed by the processor is enabled to manipulate value of individual ones of fields in a packet identifier.

56. The method of claim 39 wherein the insertion logic inserts packets into queues at least in part according to flow, and wherein insertion and selection logic cooperate to preserve order of packets in individual flows.

57. The method of claim 39 wherein the insertion logic treats packets incoming from the network or switching fabric with highest priority for identifier insertion into a queue, packet identifiers to be requeued with a next highest priority, and identifiers for packets generated by software with a lowest priority.

58. In a data packet processor having a queueing system for managing arriving packets waiting to be processed, a packet numbering system comprising:

a fixed range of packet numbers; and

an assignment facility;

characterized in that the assignment facility accounts for individual numbers in the fixed range, marking numbers invalid as packets in the queueing system are sent to be processed or software activates them, and re-assigning invalid numbers to newly arrived packets put into the queueing system.

59. The system of claim 58 wherein a probe bit is associated with each packet for ensuring that a packet number assigned to a packet at one point in time is assigned to the same packet at another point in time, characterized in that the probe bit is de-asserted when a packet number is assigned, and asserted by a Probe and Set command performed by a stream, such that at any time subsequent to the probe bit being asserted, the asserted probe bit will indicate that the packet number is associated with the same packet.